NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR VIIRS GTM IMAGERY EDR SOFTWARE (D42815 Rev A)

CDRL No. A032

Northrop Grumman Space & Mission Systems Corporation
One Space Park
Redondo Beach, California 90278

NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR VIIRS GTM IMAGERY EDR SOFTWARE (D42815 Rev A)

ELECTRONIC APPROVAL SIGNATURES:

Roy Tsugawa Date
Algorithm & Data Processing IPT Lead &
Algorithm Change Control Board Chairperson

Ben James Date
Operations & Support IPT Lead

The following individual is recognized for his contributions to the current or previous versions of this document.

Derek Stuhmer

Revision/Change Record

Document Number D42815

Revision	Document Date	Revision/Change Description	Pages Affected
---	6-30-04	Initial Release.	All
A1		22Nov05 – Raytheon-Omaha corrected for SOM versus GTM philosophy, inserted new logo, removed export markings, and worked on delta Sci2Ops code conversion process. 01Mar06 – Did edit to change title, convert document from I-Channel specific to GTM Imagery, and add M-Channel algorithm and configuration information. 14Mar06 – Did edit to correct formatting after insertion of Table 5.2-1 M-Band Unit Test dated 3/08/06 (4 pages), update to Table of Contents and List of Tables, plus copyright date on coversheet. 22Mar06 – Updated shortnames and GTM description from Peer Review actions. 29Mar06 – Updated the List of Figures to capture M-Channel Basic Processing Flow diagram, insert a TBD/TBR page, and make other minor edits. 22May06 – Added reference to EDRPR version. Updated reference to number of products produced for M-Band and fixed config guide description for B1.4. 25Oct06 – Updated OAD to include the four additional output bands. 04Jan07 – Updated OAD from GTM NCC Imagery Peer Review to convert NCC Imagery to a GTM Imagery algorithm per ECR-480.	All
A2	4-10-07	Updated wording of Detailed Algorithm description, updated to new logo.	All
A3	4-27-07	GTM FO dead detector replacement alg.	All
A4	5-3-07	Changed Table 2.2.2 to show latitude and longitude in degrees.	All
A5	5-18-07	Included updates as per NP-EMD.2006.510.0079-Rev A.	All
A6	10-29-07	Updated for code changes made under ECR-547.	All
A7	11-16-07	Changes made to OAD based on comments from Stephanie Weiss.	All
A8	11-21-07	Delivered to NGST.	All
A9	9-15-08	Reformatted to conform to the D41851 template. Updated data quality notification sections. New cover sheet, update references, acronym list, prepare for peer review. Delivered to NGST. Accept all changes after delivery.	All
A10	02-18-09	Updated with SDRL comments for TIM.	All

	NPOESS Northrop Grumman Raytheon DOC • DoD NASA

Revision/Change Record	Document Number D42815

Revision	Document Date	Revision/Change Description	Pages Affected
A	3-13-09	Incorporated TIM comments and final preparations for ARB/ACCB per ECR A-216. Approved for Public Release per Contracts Letter 100610-02.	All

Table Of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

1.1 Objective

The purpose of the Operational Algorithm Description (OAD) document is to express, in computer-science terms, the remote sensing algorithms that produce the National Polar-Orbiting Operational Environmental Satellite System (NPOESS) end-user data products. These products are individually known as Raw Data Records (RDRs), Temperature Data Records (TDRs), Sensor Data Records (SDRs) and Environmental Data Records (EDRs). In addition, any Intermediate Products (IPs) produced in the process are also described in the OAD.

The science basis of an algorithm is described in a corresponding Algorithm Theoretical Basis Document (ATBD). The OAD provides a software description of that science as implemented in the operational ground system --- the Data Processing Element (DPE).

The purpose of an OAD is two-fold:

1. Provide initial implementation design guidance to the operational software developer
2. Capture the “as-built” operational implementation of the algorithm reflecting any changes needed to meet operational performance/design requirements

An individual OAD document describes one or more algorithms used in the production of one or more data products. There is a general, but not strict, one-to-one correspondence between OAD and ATBD documents.

1.2 Scope

The scope of this document is limited to the description of the core operational algorithms required to create the VIIRS Ground Track Mercator (GTM) Imaging Band (I-Band) Imagery EDR and the VIIRS GTM Moderate Band (M-Band) Imagery EDR. The theoretical basis for these algorithms was developed by Raytheon with no ATBD as a reference document.

1.3 References

1.3.1 Document References

The science and system engineering documents relevant to the algorithms described in this OAD are listed in Table 1.

Table 1. Reference Documents

Document Title	Document Number/Revision	Revision Date
NPP EDR Production Report	D37005 Rev. C	16 Mar 2007
EDR Interdependency Report	D36385 Rev. D	18 Jun 2008
CDFCB-X Volume I - Overview	D34862-01 Rev. C	11 Jul 2008
CDFCB-X Volume II – RDR Formats	D34862-02 Rev. B	27 Aug 2007
CDFCB-X Volume III – SDR/TDR Formats	D34862-03 Rev. B	11 Jul 2008
CDFCB-X Volume IV Part 1 – IP/ARP/GEO Formats	D34862-04-01 Rev. B	07 Jul 2008
CDFCB-X Volume IV Part 2 – Atmospheric, Clouds, and Imagery EDRs	D34862-04-02 Rev. B	07 Jul 2008
CDFCB-X Volume IV Part 3 – Land and Ocean/Water EDRs	D34862-04-03 Rev. B	07 Jul 2008
CDFCB-X Volume IV Part 4 – Earth Radiation Budget EDRs	D34862-04-04 Rev. B	07 Jul 2008
CDFCB-X Volume V - Metadata	D34862-05 Rev. C	27 Jun 2008

Document Title	Document Number/Revision	Revision Date
CDFCB-X Volume VI – Ancillary Data, AuxiliaryData, Reports, and Messages	D34862-06 Rev. E	02 Jul 2008
CDFCB-X Volume VII – NPOESS Downlink Formats	D34862-07 Rev. ---	03 Jul 2008
CDFCB-X Volume VIII – Look Up Table Formats	D34862-08 Rev. ---	02 Jul 2008
NPP Mission Data Format Control Book (MDFCB)	CCR Form_429--9-02-131 Rev B	04 Apr 2006
NPP Command and Telemetry (C&T) Handbook	D568423 Rev. C	30 Sep 2008
OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR COMMON GEOLOCATION	D41869 Rev. A7	15 Sep 2008
Data Processor Inter-subsystem Interface Control Document (DPIS ICD)	D35850 Rev V	11 Feb 09
OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR VIIRS NEAR CONSTANT CONTRAST (NCC) IMAGERY EDR	D36814 Rev. A15	15 Sep 2008
D35836_E_NPOESS_Glossary	D35836_G Rev. G	10 Sep 2008
D35838_E_NPOESS_Acronyms	D35838_G Rev. G	10 Sep 2008
NGST/SE technical memo – NPP_VIIRS_GTM_Imagery_Handling_of_Bad_Detector_Data_Rev_A	NP-EMD.2006.510.0079 Rev A	16 Nov 2006

1.3.2 Source Code References

The science and operational code and associated documentation relevant to the algorithms described in this OAD are listed in Table 2. IDPS was not provided source code or test data for this algorithm. All source code and test data shown in Table 2 were developed by Raytheon.*

Table 2. Source Code References

Reference Title	Reference Tag/Revision	Revision Date
VIIRS GTM Imagery EDR Unit Test Data	*N/A	*N/A
VIIRS GTM Imagery EDR operational software	Build I1.5.x.1	Oct 2007
Procedure to bad data from VIIRS detectors in GTM Imagery	Rev. A	16 Nov 2006

2.0 ALGORITHM OVERVIEW

2.0.1 GTM Background

The purpose of the VIIRS GTM Imagery algorithm is to map VIIRS Imaging (I) channel and Moderate (M) channel data onto a GTM layout. The GTM layout is a grid of pixels, where rows are at right angles to the ground track and columns are parallel to the ground track. This GTM layout does not have the "bow-tie" effect. The GTM Imagery EDR products are primarily used for visual snow/ice analysis and to display for human viewing.

Similar to Space Oblique Mercator (SOM), the GTM is not a map projection, i.e., it does not have an exact set of unchanging transformation equations. Rather, a numerical integration process allows for a latitude and longitude calculation of a row/column (X, Y) position on the map plane, or vice versa. With SOM, there are a finite number of map planes, based on numerically integrated orbit paths. The SOM map plane for an orbit path is based on the parameters of a model orbit, followed by a numerical integration. With GTM, the actual ground track of the spacecraft establishes the map plane; consequently, the map plane is different for every orbit. The GTM map has an advantage of always having the ground track in the center of the map plane. Furthermore, multiple granules of satellite data can be concatenated without having to switch from one orbit path map to another.

2.0.2 GTM Map Description

NPOESS creates two kinds of GTM maps: Fine and Coarse. The Fine GTM map has a pixel-center spacing of 375 meters, which is close to the nadir sample distance of the VIIRS IMG resolution data. The Coarse GTM map has a pixel-center spacing of 750 meters, which is close to the nadir sample distance of the VIIRS MOD resolution data. The pixel spacing in the along track direction is equal to the pixel spacing in the cross track direction. Because of these features, the GTM map is both conformal and equal area. The maximum variation, in both conformality and area per pixel, is about one percent (a variation which is a tiny fraction smaller than the SOM map). The X-coordinate on the GTM map increases in the direction of spacecraft motion along the ground track. The X axis is precisely on the ground track. The Y axis of the map is at a right angle to the X axis. That is, the rows of the GTM map are always at an exact right angle to the ground track. The time attached to each row of the map is the time the spacecraft passes over the nadir point of that row. See Figure 1.

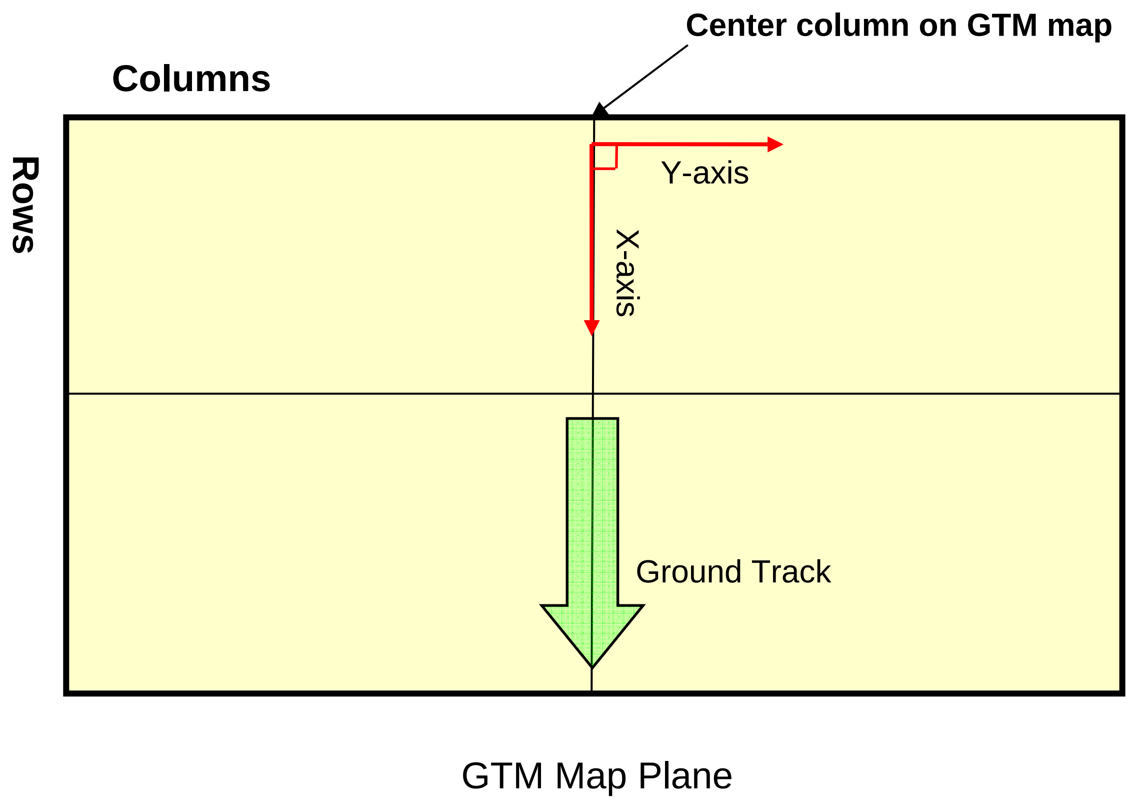

Figure 1. GTM Map Attributes

A 3090-kilometer GTM swath was chosen to accommodate a maximum satellite altitude of 850 kilometers. There are 1541 rows and 8241 columns in the fine resolution GTM layout. The row size was chosen to accommodate the minimum altitude (maximum distance of a granule) of the satellite. There can be a variable number of empty columns on the edges of the swath, due to a larger area of the Earth's surface seen near the poles and less near the equator. Rows pull together slightly at the swath edges due to Earth curvature and horizontal size of the GTM swath. There are also a variable number of empty rows at the bottom of the GTM rectangle, due to the fixed horizontal sample distance and forward ground motion of the spacecraft. This layout allows concatenation of an unlimited number of EDR granules without any discontinuities, even at the poles.

The center column of the coarse map exactly follows the center column of the fine map. The pixel centers of the coarse map center column are the same as every other pixel of the fine map center column. Similarly, the pixel centers to the left and right of the center column on the coarse map have the same centers as every other pixel on the corresponding row of the fine map. So the scaling from the coarse map to the fine map is always an exact factor of two. Once the locations of the Fine GTM map are established, the locations of the Coarse GTM map are determined by a simple 2x2 decimation. In Figure 2 each dot represents a pixel center on the GTM Fine Map. The emphasized dots represent coarse pixels on the Fine Map.

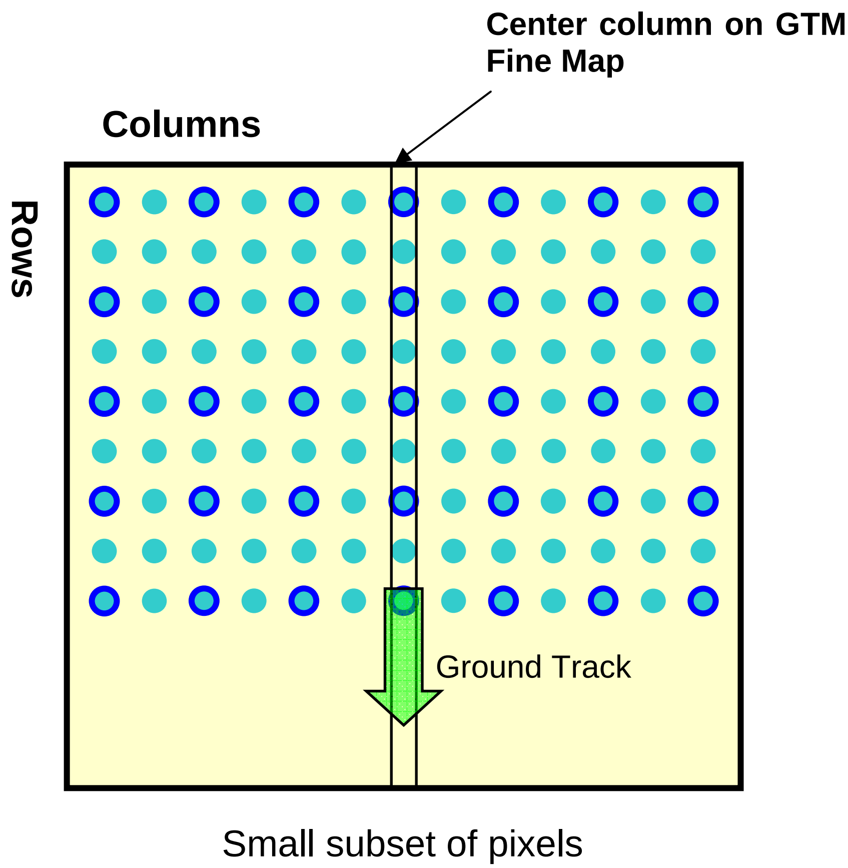

Figure 2. Fine Map Pixels with Emphasized Coarse Pixels

Even though the characteristics of the VIIRS sensor have been used to establish the parameters of the GTM maps, any kind of data can be remapped to the GTM maps. This makes it possible to form matching overlays from any number of data sources.

2.0.3 GTM Processing Overview

The first step in creating the GTM map data for an NPOESS granule is to obtain the ground track from the NPOESS Common Geolocation software. Common Geolocation functions provide the geodetic latitude, longitude, and direction of ground track motion, at least once per second. The basis of this ground track data is the ephemeris data reported by the GPS sensor on the spacecraft. This data comes down in the Ephemeris and Attitude data packets (these packets are also called "spacecraft diary" packets). Notice the direction of ground track motion is in the ECR system of the rotating Earth. This means the direction of ground track motion accounts for the rotation of the Earth as well as the inertial motion of the spacecraft. Common Geolocation is then used to calculate the ground track points for the start time and end time of the granule. Then a combination of spherical trigonometry and Common Geolocation functions are used to space the rows of the Fine GTM map as close to 375 meters as possible, and to put the center of the map precisely on the ground track. For the present granule size of about 85.752 seconds, there are about 1536 rows for Imaging resolution and 768 rows for Moderate resolution. The number of rows varies slightly because the granule size is a fixed number of seconds, and the ground track speed of the spacecraft varies slightly. The maximum variation in row spacing at nadir is 375 meters, +/- about 0.7 meters. Careful location of the first and last row in one granule means the GTM map of one granule always precisely concatenates with the GTM maps of the neighboring granules.

In summary, the centers of each row of the GTM map, the ground track pixels, are located by equal distance spacing of the pixels precisely on the ground track.

Once the locations of the ground track column pixels are established, it is possible to calculate the locations of the pixels along the rows by a simple application of spherical trigonometry. The geodetic latitude and longitude of the center pixel are used along with the radius of the Earth at that geodetic latitude. The direction from the center pixel to another pixel in the row is exactly 90 degrees to the left or right of the direction of ground track motion, which means the row is at an exact right angle to the ground track. All the pixels in one row of the map are theoretically at the same time as the center pixel, so there is no spacecraft motion or Earth rotation to account for along each row. The fact that the Earth is not an exact sphere is not a problem. The objective here is a reproducible map where there is a "one-to-one and onto" relationship between the surface of the Earth and the GTM map.

In summary, great circle distance and spherical trigonometry, location of the pixel in the center of each row, and direction of ground track motion are used to establish the location of each pixel along the row.

All of these calculations can be exactly reproduced because the entire process is based on a relatively small, single set of data: 1) the ephemeris data recorded by the GPS sensor on the spacecraft during the time span of the granule, 2) the deterministic granule boundaries (start and stop time of each granule) of the spacecraft, and 3) the 375 meter Earth surface distance between each pixel of the Fine GTM map.

2.0.4 Additional GTM Processing Details

The operational software only does full calculations for every 10th row and column, and then does quadratic interpolation of the pixels between. So, the calculation of a full set of latitudes and longitudes for a map is a relatively fast process.

The process of converting row and column to latitude and longitude, and vice versa, can be done by two methods. Method 1 is based on the fast search of a full set of geolocation data for the GTM map. Method 2 works from only the ground track data and works by an iterative search of the ground track, followed by a spherical trigonometry calculation along the row. Method 2 is slightly faster and the difference between the results is always less than one meter (the size of floating point round-off to 32 bits). The Nearest Neighbor method is used for filling pixels in order to preserve contrast and sharpness for human viewing. If full geolocation accuracy of the Sensor Data Record (SDR) is needed, the SDR should be used and not the GTM Imagery EDR.

Based on mode (day, mixed, or night) of the granule, data from either two or five imaging resolution channels are mapped onto the GTM map. In other words, two or five separate EDRs are created along with geolocation data for a given granule. Radiance and reflectance values for channels I1 through I3, along with radiance and brightness temperature values for channels I4 and I5, are processed for “day” and “mixed” mode granules. Radiance and brightness temperature values for channels I4 and I5 are processed for “night” mode granules. Currently, only six of the 16 moderate resolution channels are mapped onto the GTM map. The selection of which six channels to create is determined by a configuration file (see Section 2.3.7). It is the responsibility of the operator to select the appropriate bands for the mode of the granule day, mixed, or night) that is to be processed. Six EDRs are created using the shortnames VIIRS-M1ST-IMG-EDR through VIIRS-M6TH-IMG-EDR. The EDR metadata can be read to determine the Band ID used to generate the EDR. The Day Night Band (DNB) is processed by the Near Constant Contrast (NCC) Imagery algorithm to produce an EDR mapped to GTM. See the VIIRS NEAR CONSTANT CONTRAST IMAGERY EDR OAD, D36814, for more information. VIIRS GTM Imagery EDRs are not corrected for height.

2.1 GTM Imagery Base Class Description

2.1.1 Interfaces

2.1.1.1 Inputs

Derived classes handle all inputs.

2.1.1.2 Outputs

Derived classes handle all outputs.

2.1.2 Algorithm Processing

2.1.2.1 Main Module - ProEdrViirsGTMImagery.cpp

This is the main GTM Imagery class. It is a subclass of the ProCmnAlgorithm class and implements the methods common to all of the GTM Imagery algorithms. This class is used by the ProEdrViirsGtmIBandImagery, ProEdrViirsGtmMBandImagery, and ProEdrViirsGtmNccImagery derived classes. The inherited method applyAlgorithm() controls the flow of the GTM Imagery EDR code.

See the VIIRS NEAR CONSTANT CONTRAST (NCC) IMAGERY EDR OAD, D36814, for more information on the NCC algorithm.

2.1.2.2 createGTMImagery

All processing related routines are called from this method, and are outlined in the Program Design Language (PDL) below. This method is the main processing routine. It creates the I-Band, M-Band, or NCC Imagery products (EDRs and Geolocation dataset) for a given granule.

NOTE: The NCC Imagery part of the GTM Imagery algorithm does not actually produce the EDR. This part of the algorithm simply performs the mapping of the DNB SDR into a GTM moderate resolution sized temporary buffer. Then the NCC Imagery algorithm main routine is performed to convert the DNB radiances into the Near Constant Contrast Imagery EDR. See the NCC Imagery OAD, D36814, for more information.

Invoke the routines for creating the I-Band, M-Band, or NCC Imagery EDRs and the Geolocation EDR for a given granule.

CALL createGtmGeolocation()
CALL interpolateGridPoints()
CALL fillOutputStructures()
RETURN errorCode

2.1.2.3 createGtmGeolocation

This method calculates interpolation rectangle grid points (every 10th point in the GTM grid) and then interpolates full geolocation data for the GTM Imagery EDR using those points.

1. Calculate number of actual rows in the GTM grid, number of rows of geolocation data, and number of rows for the temporary grid data.

    a. Get granule boundary begin time and end time from an SDR.
    b. Calculate nadir point (lat/lon) at the granule begin boundary time and end boundary time.

CALL interp_ephatt(SDR ephemeris, granuleBoundaryBeginTime, &firstNadirPoint)
CALL interp_ephatt(SDR ephemeris, granuleBoundaryEndTime, &lastNadirPoint)

c. Calculate distance between the two granule boundary nadir points.
CALL azm_sidb(firstNadirPoint geodetic lat, firstNadirPoint lon, lastNadirPoint geodetic lat, lastNadirPoint lon, &azimuth, &distance)
d. Calculate radius of the earth from the average of the geodetic latitude of the first and last nadir points.
CALL earth_radius(geodetic latitude average)
e. Divide distance between granule boundaries by the horizontal sample distance, and round to the nearest integer to obtain the number of actual rows in the GTM grid.
f. Calculate number of rows for which geolocation data are generated.
g. Calculate number of rows for which temporary geolocation data are generated.

2. Calculate latitude and longitude for every 10th nadir point in the GTM grid.

   a. Calculate distance between two adjacent rows.
   b. CALL target_pt(firstNadirPoint geodetic lat, firstNadirPoint lon, distance between adjacent rows, azimuth between granule boundaries, &latitude, &longitude)

3. Calculate every 10th point to the left and right of each calculated nadir point.

   a. Refine latitude and longitude of each nadir point on the GTM grid.
   CALL refineNadirLatLon()
   b. Calculate direction for each nadir point.
   CALL calcNadirAzimuth()
   c. Add pi/2 to the direction when calculating points (lat/lons) to the left of the nadir point.
   d. Subtract pi/2 from the direction when calculating points (lat/lons) to the right of the nadir point.
   e. Calculate distance to each point left of each nadir point.
   CALL target_pt(nadir latitude, nadir longitude, distance to next point, direction, &latitude, &longitude)
   f. Calculate distance to each point to the right of each nadir point.
   CALL target_pt(nadir latitude, nadir longitude, distance to next point, direction, &latitude, &longitude)

2.1.2.4 refineNadirLatLon

This method refines the latitude and longitude of a nadir point on the GTM grid to ensure the point is on the satellite sub track.

1. Obtain geodetic latitude at the center of the primary granule from the SDR ephemeris data.

2. If this center latitude is greater than 70 degrees, then interpolate a refined nadir latitude on the basis of longitude.
   a. Search for the two longitude values in the SDR ephemeris data that surround the longitude for the given nadir point.
   b. If longitudes span the International Date Line, then adjust them so as to be working with all positive or all negative values.
   c. Linearly interpolate a refined latitude value on the basis of the nadir longitude and the two surrounding longitudes.

3. Else if the latitude is less than 70 degrees, then interpolate longitude on the basis of latitude.

a. Search for the two surrounding latitude values in the SDR ephemeris data that surround the latitude for the given nadir point.
b. If longitudes span the International Date Line, then adjust them so as to be working with all positive or all negative values.
c. Linearly interpolate a refined longitude value on the basis of the nadir latitude and the three surrounding latitudes.

2.1.2.5 calcNadirAzimuth

This method calculates azimuth for a nadir point on the GTM grid.

1. If calculating azimuth for the first nadir point, use latitude and longitude of the first nadir point and the second nadir point.

   CALL azm_sidb(latitude of first nadir point, longitude of first nadir point, latitude of second nadir point, longitude of second nadir point, &azimuth, &distance)

2. If calculating azimuth for the last nadir point, use latitude and longitude of the last nadir point and next-to-last nadir point.

   CALL azm_sidb(latitude of next-to-last nadir point, longitude of next-to-last nadir point, latitude of last nadir point, longitude of last nadir point, &azimuth, &distance)

3. If calculating azimuth for a nadir point between the first and last nadir points, use latitude and longitude of the previous nadir point and next nadir point.

   CALL azm_sidb(latitude of previous nadir point, longitude of previous nadir point, latitude of next nadir point, longitude of next nadir point, &azimuth, &distance)

2.1.2.6 interpolateGridPoints

This method interpolates GTM grid row and column values over a number of 20x20 pixel interpolation rectangles within the GTM grid.

1. Calculate every 10^{th} grid row and column value in the GTM grid using the temporary latitude and longitude values calculated previously in the createGtmGeolocation() method.

   SELECT the appropriate MDS for I-Band, M-Band, or NCC Imagery.

   CALL latlon_to_grid(latitude, longitude, granule grid mds, &grid row, &grid col)

2. Use quadratic interpolation over all 20x20 pixel interpolation rectangles to obtain all grid row and column values in the GTM grid.

   CALL quadraticInterp()

2.1.2.7 quadraticInterp

This function performs quadratic interpolation over a specified number of points.

Using three interpolation points, interpolate all values using the following formula.

   Point 1 – (x0, y0)
   Point 2 – (x1, y1)

Point 3 – (x2, y2)

c = y0
b = (y1 – y0) / (x1 – x0)
a = (((y2 – y1) / (x2 – x1)) – ((y1 – y0) / (x1 – x0))) / (x2 – x0)

p(x) = c + (b * (fx – x0)) + (a * (fx – x0) * (fx – x1))

2.1.2.8 fillOutputStructures

This virtual method fills the output geolocation and imagery EDR data structures (or the temporary SDR buffer for NCC Imagery). The derived algorithms should override this function by calling the parent class method and then copying the EDR array data.

The parent class implementation of this virtual method performs the steps in the GTM Imagery algorithm that are common to all of the derived algorithms.

CALL calculateRowTimes()
CALL calculateLatLons()
CALL the pure virtual calculateSdrPixelLocations() method for the derived class

2.1.2.8.1 fillOutputStructures in the derived algorithms

Each of the derived algorithms must override this function by calling the parent class method and then copying the EDR array data from the appropriate structures.

In general each algorithm performs the following steps:

CALL ProEdrViirsGtmImagery::fillOutputStructures()

If the return status is not PRO_SUCCESS, immediately return with the same return status.

For each output EDR perform the following steps:

Set up pointers to the Next and Previous granule data, or NULL if the Next or Previous data does not exist.

CALL the appropriate fillArray() method.

The fillArray() method is a template for the array type being copied in the EDR data (or SDR temporary buffer for NCC Imagery) and takes in as parameters the size of the data buffer being copied. The I-Band and M-Band classes use Float32 arrays and the NCC Imagery uses a Float32 array for radiance and an unsigned character array for the quality bits.

Copy the temporary pixel locations structure to the GEO output item.

This step is necessary to convert the internal array of structures into a DMS compatible structure of arrays.

2.1.2.9 calculateRowTimes

This method calculates an IET for each row in the GTM grid. The row times are written to the output geolocation structure.

1. SELECT the appropriate fine resolution geolocation structure for I-Band, M-Band, or NCC Imagery.

2. Calculate the delta time.

3. Calculate the time for each row between the first and last row in the granule using linear interpolation.

2.1.2.10 calculateLatLons

This method calculates the latitude and longitude for each pixel in the GTM granule and fills appropriate fields in the output geolocation structure. In the NCC and M-Band processes, the temporary fine resolution geolocation structure is filled.

1. SELECT the appropriate MDS and fine resolution geolocation structure for I-Band, M-Band, or NCC Imagery.

2. Calculate the latitude and longitude for each grid point and fill the output imagery EDR geolocation data structure with these values. It is preferred to only loop over the number of actual GTM rows in the temporary grid row and column arrays.

   CALL grid_to_latlon(grid row, grid column, granule grid MDS, &latitude, &longitude)

2.1.2.11 calculateSdrPixelLocations

This pure virtual method is used to calculate the corresponding SDR pixel location for each pixel in the Imaging resolution GTM granule. The derived algorithms must override this function as described below.

2.1.2.12 calculateSdrPixelLocations in the derived algorithms

This method calculates the corresponding SDR pixel location for each pixel in the Imaging resolution GTM granule. These values are written to a temporary array and are later written to the output geolocation structure.

This method is a pure virtual method that must be instantiated by each of the derived algorithms to operate on the appropriately sided data structures. The M-Band and NCC Imagery version of this method should decimate the fine resolution geolocation into a coarse resolution geolocation structure.

1. Create two grid point conversion objects for converting polar stereographic grid points between the primary granule's grid and a neighboring granule's grid.

2. In the NCC and M-Band processes, use 2x2 decimation to reduce the temporary fine resolution geolocation structure to the coarse resolution output structure. Also decimate the fine resolution Grid Row Column structure to a temporary coarse resolution structure.

3. Loop over 20x20 pixel rectangles in the GTM Imagery EDR geolocation data.

   a. Calculate distance between two columns at the center of a 20x20 rectangle.
   b. CALL azm_sidb()
   c. Calculate distance between two rows at the center for a 20x20 rectangle.
   d. CALL azm_sidb()

e. Loop over all GTM grid points in the current 20x20 rectangle and convert each grid point to an SDR pixel location.

CALL the appropriate grid to SDR pixel method:

grid_to_modSDRpixel(grid row, grid column, &pixelRow, &pixelCol)
grid_to_imgSDRpixel(grid row, grid column, &pixelRow, &pixelCol)
grid_to_dnbSDRpixel(grid row, grid column, &pixelRow, &pixelCol)

If the method returns an error (pixel search failed or not a valid warning code), then send a debug message and return PRO_FAIL.

If the method returns a warning (pixel not in the SDR), then

1. Map the pixel based on the errorCode using the appropriate grid point conversion object created above.
2. CALL the appropriate grid to SDR pixel method again.
3. Fill that pixel with a flag value based on the second errorCode returned by the grid to SDR pixel method.

If the pixel is in the primary SDR granule, then

1. Check for boundary row pixel trimmed values and convert them to the previous or next granule. Otherwise, the pixel is in the primary granule and the row and column have been properly calculated.
2. Store the pixel location.

2.1.2.13 Reuse C Functions

2.1.2.13.1 ProSdrCmnGeo::setupEphatt

See the Common Geolocation OAD, D41869, for a description of this routine.

2.1.2.13.2 ProSdrCmnGeo::satPosAtt

See the Common Geolocation OAD, D41869, for a description of this routine.

NOTE: The points in the ephatt structure are spaced one second apart, making linear interpolation acceptable.

2.1.2.13.3 azm_sidb

This function calculates bearing (in radians) and distance (earth central angle in radians) from point C to point A, based on C and A latitude/longitude of points. Table 3 shows the azm_sidb parameter definitions.

Table 3. azm_sidb Parameter Definitions

Parameter	Type	I/O	Description
Clat	Double	I	Latitude of the first point, in radians positive north.
Clon	Double	I	Longitude of the first point, in radians positive east.
Alat	Double	I	Latitude of the second point, in radians positive north.
Alon	Double	I	Longitude of the second point, in radians positive east.
Azimuth	double*	O	Bearing from C to A, in radians.
side_b	double*	O	Distance from C to A, in radians.

NOTE: "Side_b" is a distance measured as an angle at the center of the earth. The angle goes from point C to point A, along the great circle between, and the vertex is the center of the earth. The maximum value of "side_b" is pi.

NOTE: "Bearing" is the standard definition. "Bearing" in radians goes from zero to 2pi, with North=0, East=pio2, South=pi, West=trepio2. "Bearing" and "azimuth" are the same thing.

CONSTRAINTS: All inputs for latitude and longitude are positive north positive east.
-pio2 < latitude < pio2, -pi < longitude < pi
Making sure the inputs are in a legal range is the responsibility of the calling program.

2.1.2.13.4 target_pt

Below is a discussion of function target_pt, which does all of the spherical trig calculations described in Section 2.0, Algorithm Overview. Figure 3 shows a diagram of how these calculations are made.

The spherical trigonometry used to calculate a second point from a first point is based on latitude and longitude of the start point, radius of the sphere at the start point, direction from start point to target point, and the laws of spherical trigonometry. The oblique spherical triangle used has the vertices: vertex A is the target point (unknown lat/lon), vertex B is the North Pole, and vertex C is the start point.

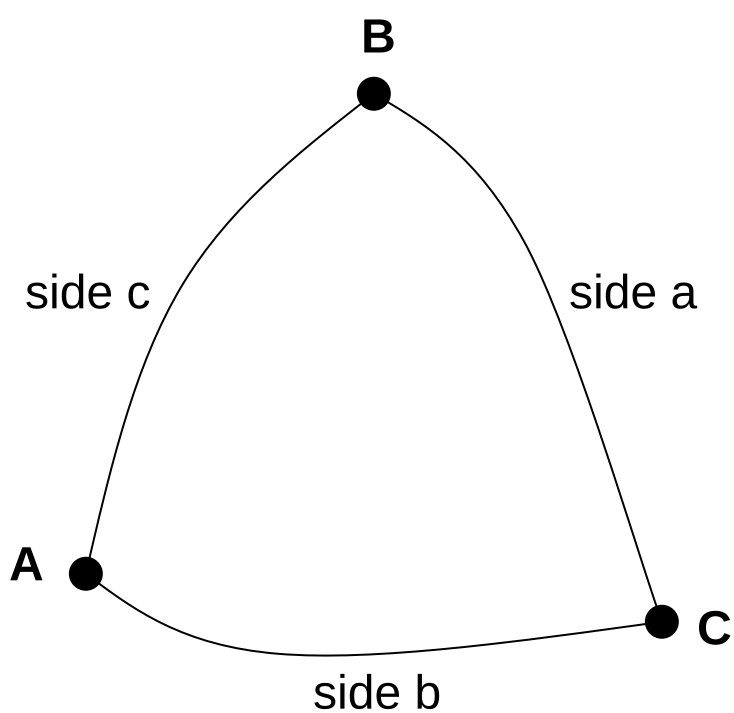

Figure 3. Target_pt Function Calculations Diagram

In spherical trigonometry, each side of the triangle is a great circle arc between the two vertices. The length of the side of a triangle is the angle measured at the center of the sphere, measured along the great circle arc from one vertex to the other.

From the input azimuth, it is determined that the target point is East or West of the start point. It is also known that side b is the distance from start point to target point, divided by the radius of the Earth at the start point.

Side a of this triangle is the longitude line from the North Pole to the start point. So, side a is equal to 90 degrees minus the latitude of the start point.

Angle C is directly determined by the azimuth from the start point to the target point. Angle C has to be adjusted depending on whether the target point is East or West of the start point.

Then the cosine of side c is determined from the Law of Cosines for Oblique Spherical triangles:

$$\cos(c) = \cos(a) \bullet \cos(b) + \sin(a) \bullet \sin(b) \bullet \cos(C)$$

Then side c is determined from the arc cosine function, and the latitude of the target point (one output of the function) is just (90 - side c).

Then the cosine of angle B is determined by again applying and rearranging the Law of Cosines for Oblique Spherical Triangles:

$$\cos(B) = \frac{\cos(b) - \cos(c) \bullet \cos(a)}{\sin(c) \bullet \sin(a)}$$

Then angle B is determined from the arc cosine function and the longitude of the target point (the second output of the function) is the start point longitude plus or minus angle B, depending whether the target point is East or West of the start point.

This function has the inputs of a latitude and longitude start point, plus distance and direction to a second point. Outputs are latitude and longitude of the second point. Table 4 shows the target_pt parameter definitions.

Table 4. target_pt Parameter Definitions

Parameter	Type	I/O	Description
Clat	Double	I	Latitude of the start point.
Clon	Double	I	Longitude of the start point.
side_b	Double	I	Distance to the second point.
Azimuth	Double	I	Azimuth to the second point.
Alat	Double*	O	Latitude of the second point.
Alon	Double*	O	Longitude of the second point.

NOTE: All latitudes and longitudes are in radians, positive North and positive East.
-pio2 < Latitude < pio2, -pi < Longitude < pi

NOTE: side_b is an angle measured at the center of the earth.

CONSTRAINTS: It is the responsibility of the calling program to keep inputs inside legal ranges.

NOTE: This function resolves a triangle on the surface of a sphere using laws from spherical trigonometry. Vertices of the triangle are: Point A (target point), Point B (North Pole), and Point C (start point).

2.1.2.13.5 grid_to_imgSDRpixel

This function takes in an interpolation grid position and the geolocation data for an Imaging Resolution SDR granule, and then converts location to an SDR pixel location (row and column). Table 5 shows the grid_to_imgSDRpixel parameter definitions.

Table 5. grid_to_imgSDRpixel Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	Input row on granule interpolation grid.
gcol	Double	I	Input column on granule interpolation grid.
iRect	ViirsGeoRctnglType&	I	Rectangle parameters.
vI_growcol	viirs_SDR_IMG_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.13.6 grid_to_modSDRpixel

This function takes in an interpolation grid position and the geolocation data for a Moderate Resolution SDR granule, and then converts location to an SDR pixel location (row and column). Table 6 shows the grid_to_modSDRpixel parameter definitions.

Table 6. grid_to_modSDRpixel Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	Input row on granule interpolation grid.
gcol	Double	I	Input column on granule interpolation grid.
iRect	ViirsGeoRctnglType&	I	Rectangle parameters.
vM_growcol	viirs_SDR_MOD_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.13.7 grid_to_dnbSDRpixel

This function takes in an interpolation grid position and the geolocation data for a Day Night Band SDR granule, and then converts location to an SDR pixel location (row and column). Table 7 shows the grid_to_imgSDRpixel parameter definitions.

Table 7. grid_to_imgSDRpixel Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	Input row on granule interpolation grid.
gcol	Double	I	Input column on granule interpolation grid.
iRect	ViirsGeoRctnglType&	I	Rectangle parameters.
vD_growcol	viirs_SDR_DNB_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.13.8 rp_g2imgpix

This function refines pixel location of the input grid position in the VIIRS IMG granule. Function grid_to_imgSDRpixel takes in grow,gcol and determines the pixel (iprow,ipcol) closest to that input. This function refines that location to a fraction of a pixel. Table 8 shows the rp_g2imgpix parameter definitions.

Table 8. rp_g2imgpix Parameter Definitions

Parameter	Type	I/O	Description
Grow	Double	I	Input row on granule interpolation grid.
Gcol	Double	I	Input column on granule interpolation grid.
Iprow	Int	I	Row of pixel closest to input point.
Ipcol	Int	I	Column of pixel closest to input point.
Iprow_bgn	Int	I	First row of scan containing iprow.
Iprow_end	Int	I	Last row of scan containing iprow.
vI_growcol	viirs_SDR_IMG_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.13.9 rp_g2modpix

This function refines pixel location of the input grid position in the VIIRS MOD granule. Function grid_to_modSDRpixel takes in grow,gcol and determines the pixel (iprow,ipcol) closest to that input. This function refines that location to a fraction of a pixel. Table 9 shows the rp_g2modpix parameter definitions.

Table 9. rp_g2modpix Parameter Definitions

Parameter	Type	I/O	Description
Grow	Double	I	Input row on granule interpolation grid.
Gcol	Double	I	Input column on granule interpolation grid.
Iprow	Int	I	Row of pixel closest to input point.
Ipcol	Int	I	Column of pixel closest to input point.
Iprow_bgn	Int	I	First row of scan containing iprow.
Iprow_end	Int	I	Last row of scan containing iprow.
vM_growcol	viirs_SDR_MOD_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.13.10 rp_g2dnbdpix

This function refines pixel location of the input grid position in the VIIRS DNB granule. Function grid_to_dnbSDRpixel takes in grow,gcol and determines the pixel (iprow,ipcol) closest to that input. This function refines that location to a fraction of a pixel. Table 10 shows the rp_g2dnbdpix parameter definitions.

Table 10. rp_g2dnbdpix Parameter Definitions

Parameter	Type	I/O	Description
Grow	Double	I	Input row on granule interpolation grid.

Parameter	Type	I/O	Description
Gcol	Double	I	Input column on granule interpolation grid.
Iprow	Int	I	Row of pixel closest to input point.
Ipcol	Int	I	Column of pixel closest to input point.
Iprow_bgn	Int	I	First row of scan containing iprow.
Iprow_end	Int	I	Last row of scan containing iprow.
vD_growcol	viirs_SDR_DNB_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.13.11 grid_to_latlon

This function converts a row column position from an MDS to a latitude and longitude. Table 11 shows the grid_to_latlon parameter definitions.

Table 11. grid_to_latlon Parameter Definitions

Parameter	Type	I/O	Description
Row	Double	I	Row coordinate on the grid.
Col	Double	I	Column coordinate on the grid.
Imds	mds_type*	I	Pointer to the input map data set structure.
Rlat	double*	O	Pointer to the output latitude.
Rlon	double*	O	Pointer to the output longitude.
err_string	char*	O	Pointer to a 256 byte string.
Err	Int	O	Returned error code.

2.1.2.13.12 latlon_to_grid

This function converts a latitude and longitude to a position on an MDS grid. Table 12 shows the latlon_to_grid parameter definitions.

Table 12. latlon_to_grid Parameter Definitions

Parameter	Type	I/O	Description
Rlat	Double	I	Input latitude.
Rlon	Double	I	Input longitude.
Imds	mds_type*	I	Pointer to the input map data set structure.
Row	double*	O	Pointer to the row coordinate on the grid.
Col	double*	O	Pointer to the column coordinate on the grid.
Err_string	char*	O	Pointer to a 256 byte string.
Err	Int	O	Returned error code.

2.1.2.13.13 earth_radius_D

This function computes radius of the Earth, in kilometers, from the geodetic latitude. Table 13 shows the earth_radius_D parameter definitions.

Table 13. earth_radius_D Parameter Definitions

Parameter	Type	I/O	Description
Rlat	Double	I	Geodetic latitude, radians, positive north.
Radius	Double	O	Returned radius of the earth in kilometers.

2.1.3 Graceful Degradation

2.1.3.1 Graceful Degradation Inputs

There is one case where input graceful degradation is indicated in the VIIRS GTM Imagery EDR.

1. An input retrieved for the algorithm had its N_Graceful_Degradation metadata field set to YES (propagation).

2.1.3.2 Graceful Degradation Processing

None.

2.1.3.3 Graceful Degradation Outputs

None.

2.1.4 Exception Handling

Missing sensor data caused by bad detectors are replaced during pre-processing of the GTM algorithm as follows:

For edge of scan, the radiance value of the adjacent pixel is copied into the missing pixel. For non-edge of scan, the radiances are averaged using the two adjacent pixels and copied into the missing pixel. Missing data points are left as the initialization fill value in the GTM Imagery EDRs.

Additionally, the VIIRS GTM Imagery software is designed to handle a wide variety of processing problems. Any exceptions or errors are reported to IDPS using the appropriate INF API.

2.1.5 Data Quality Monitoring

Data quality monitoring is left to the derived classes of this base class.

2.1.6 Computational Precision Requirements

The GTM Imagery algorithm is a pass-through of the SDR radiance, reflectance, and brightness temperature data, and no change of precision occurs in the code. Geolocation (latitude and longitude) computations are done in 32-bit and 64-bit floating point precision. The final latitude and longitude values are accurate to one meter. All time computations are done in 64-bit floating point and 64-bit integer precision.

In order to speed GTM Imagery processing, an interpolation scheme is used to calculate grid data for the granule. The latitude and longitude and associated grid row and column values are calculated for every 10^{th} point in the GTM grid. Then quadratic interpolation is performed over 20x20 pixel rectangles within the GTM grid to obtain the full geolocation for the granule. The maximum error induced by the interpolation is one meter. Row times are calculated by performing linear interpolation between the granule boundary times.

2.1.7 Algorithm Support Considerations

DMS and INF must be running before execution of the GTM Imagery algorithm.

2.1.8 Assumptions and Limitations

No assumptions or limitations have been identified.

2.2 GTM Imagery I-Band Class Description

This is the main GTM I-Band Imagery process. It instantiates an I-Band instance of the ProEdrViirsGTMImagery class, which is a subclass of the ProEdrViirsImagery class, and calls the applyAlgorithm() method.

The basic flow of the I-Band Imagery algorithm is depicted in Figure 4. Inputs are the VIIRS SDRs (channels I1 through I5), VIIRS I-Band SDR grid data, and VIIRS I-Band sensor look angles. Outputs are the I-Band Imagery EDRs and geolocation data.

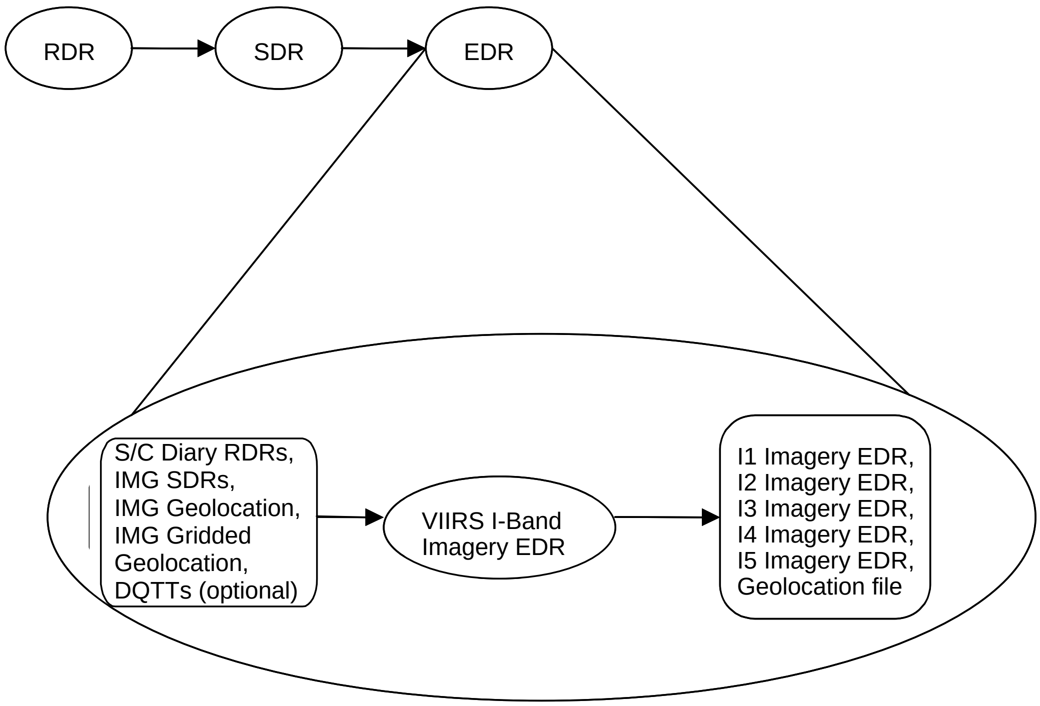

Figure 4. Basic Processing Flow for the VIIRS I-Band Imagery EDR

2.2.1 Interfaces

2.2.1.1 Inputs

VIIRS I-Band Imagery algorithm requires several types of data to perform mapping to the GTM layout, summarized in Table 14. Refer to the CDFCB-X for a detailed description of the inputs.

Table 14. VIIRS I-Band Imagery EDR Inputs

Name	Description
Spacecraft Diary RDRs	Attitude and ephemeris data for the granule.
VIIRS Band I1, I2, and I3 SDRs	Radiances and reflectances, granule boundary times and granule mode. Used for day and mixed mode granules.
VIIRS Band I4 and I5 SDRs	Brightness temperatures and radiances, granule boundary times and granule mode. Used for day, mixed and night mode granules.
VIIRS IMG Geolocation	Ellipsoid Geolocation data for every pixel in the granule.
VIIRS IMG Gridded Geolocation	Map grid row and column values for every pixel in the granule and the granule MDS.
Band I1–I5 EDR DQTTs	Data Quality Threshold Tables used for performing data quality checks on the EDR outputs. These are optional inputs.

2.2.1.2 Outputs

VIIRS I-Band Imagery EDRs are summarized in Table 15. Note that the I1 – I3 band EDRs have a reflectance field where as the I4 and I5 band EDRs have a brightness temperature field. The VIIRS I-Band Imagery GEO is summarized in Table 17. Latitude and longitude are calculated as the center of the GTM pixel. Solar angle, satellite angle, terrain height and satellite range are copied from the source SDR pixel. Refer to the CDFCB-X for a detailed description of the outputs.

Table 15. VIIRS I-Band Imagery EDR Outputs

Output	Type	Description	Units / Range
radiance	Uint16 * [1541 * 8241]	Top of Atmosphere radiances for the I1-I5 Bands	W/(m2·sr·µm) / -1.2 – 87.0
reflectance (I1 – I3 EDRs only)	Uint16 * [1541 * 8241]	Top of Atmosphere Reflectances (Daytime only) for the I1 - I3-Bands	Unitless / 0.0 – 1.6
brightTemp (I4, I5 EDRs only)	Uint16 * [1541 * 8241]	Top of Atmosphere Equivalent Blackbody Brightness Temperatures for the I4, I5-Bands	Degree / 208 - 367
pixelQuality	Uint8 * [1541 * 8241]	Pixel-level quality flags	See Table 16
radScale	Float32	Scale for scaled radiance	Unitless
radOff	Float32	Offset for scaled radiance	Unitless
refl/btScale	Float32	Scale for Reflectance / Brightness Temp	Unitless
refl/btOff	Float32	Offset for Reflectance / Brightness Temp	Unitless

Table 16. VIIRS I-Band Imagery EDR Pixel Level Quality Flags

Byte	Bits	Description	Values
Byte 0	0-1	Imagery Quality (Pixel Quality as determined by the SDR Calibration Quality. Dead Pixel Replacement: Individual bad pixels caused by a bad detector are filled as an average of the two adjacent detector pixels. Bad edge-of-scan pixels use the adjacent pixel value. (If two adjacent pixels are dead, a fill value is used for each pixel.)	0: Good 1: Poor 2: No Calibration 3: Dead pixel replacement
	2	Pixel is saturated	0: False 1: True
	3-4	Missing Data (Data required for calibration processing is not available for processing)	0: All data present 1: EV missing 2: Cal data missing 3: Therm data missing
	5-6	Out of range	0: Good 1: Radiance out of range 2: Reflectance or EBBT out of range 3: Both out of range
	7	Spare	

Table 17. VIIRS I-Band Imagery GEO Outputs

Output	Type	Description	Units / Range
rowTime	Int64 * [1541]	Time of the nadir point of the GTM row in IET (1/1/1958). Represents the time of the nadir point of the GTM row.	Microsecond / 1483228832000000 - 2272147232000000
Latitude	Float32 * [1541 * 8241]	Latitude of each pixel (positive North) Calculated as the center of the GTM pixel	Degree / -90 - 90
Longitude	Float32 * [1541 * 8241]	Longitude of each pixel (positive East) Calculated as the center of the GTM pixel	Degree / -180 - 180
sunZenith	Float32 * [1541 * 8241]	Zenith angle of sun at each pixel position Copied from the source SDR pixel	Degree / 0 - 180
sunAzimuth	Float32 * [1541 * 8241]	Azimuth angle of sun (measured clockwise positive from North) at each pixel position	Degree / -180 - 180

Output	Type	Description	Units / Range
sensorZenith	Float32 * [1541 * 8241]	Zenith angle to Satellite at each pixel position Copied from the source SDR pixel	Degree / 0 - 180
sensorAzimuth	Float32 * [1541 * 8241]	Azimuth angle (measured clockwise positive from North) to Satellite at each pixel position Copied from the source SDR pixel	Degree / -180 - 180
terrainHeight	Int16 * [1541 * 8241]	Ellipsoid-Geoid separation Copied from the source SDR pixel	Meter / -500 - 8300
satRange	Float32 * [1541 * 8241]	Line of sight distance from the ellipsoid intersection to the satellite Copied from the source SDR pixel	Meter / 820,000 -1,900,000
pixelQuality	Uint8 * [1541 * 8241]	Pixel Level Geolocation Quality Flags	Refer to Table 18
scanQuality	Uint8 * [48]	Scan Level Geolocation Quality Flags	Refer to Table 19
sdrRow	UInt16 * [1541 * 8241]	Imagery SDR pixel row index number that was remapped to this GTM pixel (row numbering begins with zero)	Unitless
sdrCol	UInt16 * [1541 * 8241]	Imagery SDR pixel column index number that was remapped to this GTM pixel (column numbering begins with zero)	Unitless

Table 18. VIIRS I-Band Imagery GEO Pixel Level Quality Flags

Byte	Bits	Description	Values
Byte 0	0-1	SDR Pixel Mapping Coordinate (GTM to SDR). Indicates whether this pixel originated from the previous, current, or next granule in the SDR Imagery Resolution Geolocation.	0: Error 1: Previous Granule 2: Current Granule 3: Next Granule
	2-7	Spare	

Table 19. VIIRS I-Band Imagery GEO Scan Level Quality Flags

Byte	Bits	Description	Values
Byte 0	0	Solar Eclipse	0: No Solar Eclipse 1: Solar Eclipse
	1-7	Spare	

2.2.2 Algorithm Processing

2.2.2.1 Main Module - ProEdrViirsGtmIBandImagery.cpp

This class is the implementation of the VIIRS Imaging Band Imagery algorithm that computes the I-Band Imagery EDRs mapped to the Ground Track Mercator (GTM) map. This class inherits from ProEdrViirsGtmImagery.

The I-Band Imagery EDRs contain VIIRS imaging band data mapped onto a Ground Track Mercator (GTM) layout. Depending on the granule mode (DAY, MIXED, or NIGHT), two or five I-Band Imagery EDRs are created per granule. One geolocation dataset is also created per granule.

2.2.2.2 setupDataItems

This method implements the pure virtual base class method. It creates all the input and output data items needed for VIIRS I-Band Imagery processing.

2.2.2.3 doProcessing

The algorithm does a pre-processing step using the 32-bit bad detector quality flags located in the SDR input. If the first detector is bad, the radiance value in [row+1] is copied into the current row. If the last detector is bad, the radiance value in [row-1] is copied into the current row. For other bad detectors, the before and after rows are averaged and placed into the current row.

2.2.2.4 initOutputDataItems

This method initializes each output data item's DMS data buffer.

2.2.3 Graceful Degradation

2.2.3.1 Graceful Degradation Inputs

There is one case where input graceful degradation is indicated in the VIIRS GTM Imagery EDR.

1. An input retrieved for the algorithm had its N_Graceful_Degradation metadata field set to YES (propagation).

2.2.3.2 Graceful Degradation Processing

None.

2.2.3.3 Graceful Degradation Outputs

None.

2.2.4 Exception Handling

Missing sensor data caused by bad detectors are replaced during pre-processing of the GTM algorithm as follows:

For edge of scan, the radiance value of the adjacent pixel is copied into the missing pixel. For non-edge of scan, the radiances are averaged using the two adjacent pixels and copied into the missing pixel. Missing data points are left as the initialization fill value in the GTM Imagery EDRs.

Additionally, the VIIRS GTM Imagery software is designed to handle a wide variety of processing problems. Any exceptions or errors are reported to IDPS using the appropriate INF API.

2.2.5 Data Quality Monitoring

Each algorithm uses specific criteria contained in a Data Quality Threshold Table (DQTT) to determine when a Data Quality Notification (DQN) is produced. The DQTT contains the threshold used to trigger the DQN as well as the text contained in the DQN. If a threshold is met, the algorithm stores a DQN in DMS indicating the test(s) that failed and the value of the DQN attribute. For more algorithm specific detail refer to the CDFCB-X, D34862.

2.2.6 Computational Precision Requirements

The GTM Imagery algorithm is a pass-through of the SDR radiance, reflectance, and brightness temperature data, and no change of precision occurs in the code. Geolocation (latitude and longitude) computations are done in 32-bit and 64-bit floating point precision. The final latitude and longitude values are accurate to one meter. All time computations are done in 64-bit floating point and 64-bit integer precision.

In order to speed GTM Imagery processing, an interpolation scheme is used to calculate grid data for the granule. The latitude and longitude and associated grid row and column values are calculated for every 10th point in the GTM grid. Then quadratic interpolation is performed over 20x20 pixel rectangles within the GTM grid to obtain the full geolocation for the granule. The maximum error induced by the interpolation is one meter. Row times are calculated by performing linear interpolation between the granule boundary times.

2.2.7 Algorithm Support Considerations

DMS and INF must be running before execution of the GTM Imagery algorithm.

2.2.8 Assumptions and Limitations

No assumptions or limitations have been identified.

2.3 GTM Imagery M-Band Class Description

This is the main GTM M-Band Imagery process. It instantiates an M-Band instance of the ProEdrViirsGTMImagery class, which is a subclass of the ProEdrViirsImagery class, and calls the applyAlgorithm() method.

The basic flow of the M-Band Imagery algorithm is depicted in Figure 5. Inputs are the VIIRS Mod SDRs, VIIRS Mod SDR grid data, and Mod VIIRS sensor look angles. Outputs are the M-Band Imagery EDRs and geolocation data.

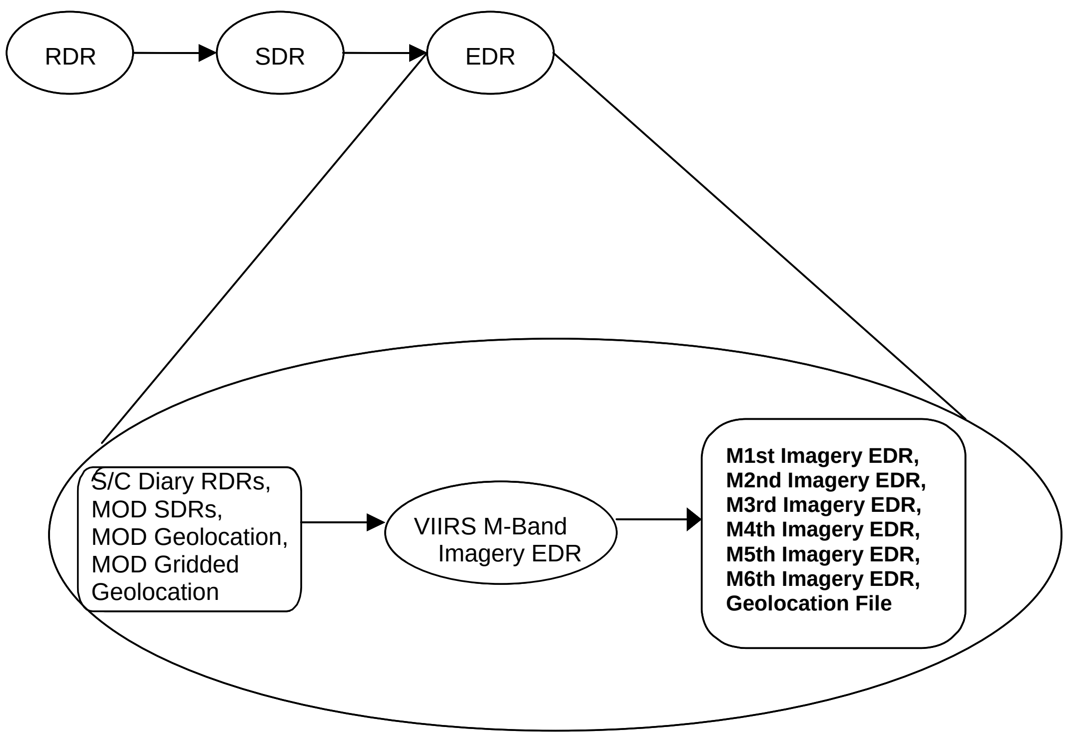

Figure 5. Basic Processing Flow for the VIIRS M-Band Imagery EDR

2.3.1 Interfaces

2.3.1.1 Inputs

VIIRS M-Band Imagery algorithm requires several types of data to perform mapping to the GTM layout, summarized in Table 20. Refer to the CDFCB-X for a detailed description of the inputs.

Table 20. VIIRS M-Band Imagery EDR Inputs

Name	Description
Spacecraft Diary RDRs	Attitude and ephemeris data for the granule.
VIIRS Band M1 thru M6, M9, and M11 SDRs	Radiances and reflectances, granule boundary times and granule mode. Used for day and mixed mode granules.
VIIRS Band M7, M8, M10, and M12 thru M16 SDRs	Radiances and brightness temperatures, granule boundary times and granule mode. Used for day, mixed and night mode granules.
VIIRS MOD Geolocation	Ellipsoid Geolocation data for every pixel in the granule.
VIIRS MOD Gridded Geolocation	Map grid row and column values for every pixel in the granule and the granule MDS.

2.3.1.2 Outputs

VIIRS M-Band Imagery EDRs are summarized in Table 21. Note that the M1 – M6, M9, and M11 band EDRs have a reflectance field where as the M7, M8, M10, and M12 – M16 band EDRs have a brightness temperature field. The VIIRS M-Band Imagery GEO is summarized in Table 22. Latitude and longitude are calculated as the center of the GTM pixel. Solar angle, satellite angle, terrain height and satellite range are copied from the source SDR pixel. Refer to the CDFCB-X for a detailed description of the outputs.

Table 21. VIIRS M-Band Imagery EDR Outputs

Output	Type	Description	Units / Range
radiance	Uint16 * [771 * 4121]	TOA radiances for the M-Band selected	W/(m2·sr·µm) / -30.0 – 842.4
reflectance (M1 – M6, M9, M11 EDRs only)	Uint16 * [771 * 4121]	Top of Atmosphere Reflectances	Unitless / 0.0 – 1.6
brightTemp (M7, M8, M10, M12 – M16 EDRs only)	Uint16 * [771 * 4121]	Top of Atmosphere Equivalent Blackbody Brightness Temperature	Degree / 192.0 – 683.0
radScale	Float32	Scale for scaled radiance	Unitless
radOff	Float32	Offset for scaled radiance	Unitless
refl/btScale	Float32	Scale for Reflectance / Brightness Temp	Unitless
refl/btOff	Float32	Offset for Reflectance / Brightness Temp	Unitless

Table 22. VIIRS M-Band Imagery GEO Output

Output	Type	Description	Units/Range
rowTime	Int64 * [771]	Time of the nadir point of the GTM row in IET (1/1/1958). Represents the time of the nadir point of the GTM row	Microsecond / 1483228832000000 - 2272147232000000
Latitude	Float32 * [771 * 4121]	Latitude of each pixel (positive North) Calculated as the center of the GTM pixel	Degree / -90 - 90
Longitude	Float32 * [771 * 4121]	Longitude of each pixel (positive East) Calculated as the center of the GTM pixel	Degree / -180 - 180
sunZenith	Float32 * [771 * 4121]	Zenith angle of sun at each pixel position Copied from the source SDR pixel	Degree / 0 - 180

Output	Type	Description	Units/Range
sunAzimuth	Float32 * [771 * 4121]	Azimuth angle of sun (measured clockwise positive from North) at each pixel position Copied from the source SDR pixel	Degree / -180 - 180
sensorZenith	Float32 * [771 * 4121]	Zenith angle to Satellite at each pixel position Copied from the source SDR pixel	Degree / 0 - 180
sensorAzimuth	Float32 * [771 * 4121]	Azimuth angle (measured clockwise positive from North) to Satellite at each pixel position Copied from the source SDR pixel	Degree / -180 - 180
terrainHeight	Int16 * [771 * 4121]	Ellipsoid-Geoid separation Copied from the source SDR pixel	Meter / -500 - 8300
satRange	Float32 * [771 * 4121]	Line of sight distance from the ellipsoid intersection to the satellite Copied from the source SDR pixel	Meter / 820,000 – 1,900,000
pixelQuality	Uint8 * [771 * 4121]	Pixel Level Geolocation Quality Flags	Refer to Table 23
granuleQualFlag	Uint8	Granule Level Geolocation Quality Flags	Refer to Table 24
sdrRow	UInt16 * [771 * 4121]	Moderate SDR pixel row index number that was remapped to this GTM pixel (row numbering begins with zero)	Unitless
sdrCol	UInt16 * [771 * 4121]	Moderate SDR pixel column index number that was remapped to this GTM pixel (column numbering begins with zero)	Unitless

Table 23. VIIRS M-Band Imagery GEO Pixel Level Quality Flags

Byte	Bits	Description	Values
Byte 0	0-1	SDR Pixel Mapping Coordinate (GTM to SDR). Indicates whether this pixel originated from the previous, current, or next granule in the SDR Imagery Resolution Geolocation.	0: Error 1: Previous Granule 2: Current Granule 3: Next Granule
	2-7	Spare	

Table 24. VIIRS M-Band Imagery GEO Granule Level Quality Flag

Byte	Bits	Description	Values
Byte 0	0	Solar Eclipse	0: No Solar Eclipse 1: Solar Eclipse
	1-7	Spare	

2.3.2 Algorithm Processing

2.3.2.1 Main Module - ProEdrViirsGtmMBandImagery.cpp

This class is the implementation of the VIIRS Moderate Band Imagery algorithm that computes the M-Band Imagery EDRs mapped to the Ground Track Mercator (GTM) map. This class inherits from ProEdrViirsGtmImagery.

The M-Band Imagery EDRs contain VIIRS moderate band data mapped onto a Ground Track Mercator (GTM) layout. Depending on the keys selected in the configuration guide, six out of the 16 M-Band Imagery EDRs are created per granule. One geolocation dataset is also created per granule.

2.3.2.2 setupDataItems

This method implements the pure virtual base class method. It creates all the input and output data items needed for VIIRS M-Band Imagery processing.

2.3.2.3 doProcessing

The algorithm does a pre-processing step using the 16-bit bad detector quality flags located in the SDR input. If the first detector is bad, the radiance value in [row+1] is copied into the current row. If the last detector is bad, the radiance value in [row-1] is copied into the current row. For other bad detectors, the before and after rows are averaged and placed into the current row.

2.3.2.4 initOutputDataItems

This method initializes each output data item's DMS data buffer.

2.3.3 Graceful Degradation

2.3.3.1 Graceful Degradation Inputs

There is one case where input graceful degradation is indicated in the VIIRS GTM Imagery EDR.

1. An input retrieved for the algorithm had its N_Graceful_Degradation metadata field set to YES (propagation).

2.3.3.2 Graceful Degradation Processing

None.

2.3.3.3 Graceful Degradation Outputs

None.

2.3.4 Exception Handling

Missing sensor data caused by bad detectors are replaced during pre-processing of the GTM algorithm as follows:

For edge of scan, the radiance value of the adjacent pixel is copied into the missing pixel. For non-edge of scan, the radiances are averaged using the two adjacent pixels and copied into the missing pixel. Missing data points are left as the initialization fill value in the GTM Imagery EDRs.

Additionally, the VIIRS GTM Imagery software is designed to handle a wide variety of processing problems. Any exceptions or errors are reported to IDPS using the appropriate INF API.

2.3.5 Data Quality Monitoring

Each algorithm uses specific criteria contained in a Data Quality Threshold Table (DQTT) to determine when a Data Quality Notification (DQN) is produced. The DQTT contains the threshold used to trigger the DQN as well as the text contained in the DQN. If a threshold is met, the algorithm stores a DQN in DMS indicating the test(s) that failed and the value of the DQN attribute. For more algorithm specific detail refer to the CDFCB-X, D34862.

2.3.6 Computational Precision Requirements

The GTM Imagery algorithm is a pass-through of the SDR radiance, reflectance, and brightness temperature data, and no change of precision occurs in the code. Geolocation (latitude and longitude) computations are done in 32-bit and 64-bit floating point precision. The final latitude and longitude values are accurate to one meter. All time computations are done in 64-bit floating point and 64-bit integer precision.

In order to speed GTM Imagery processing, an interpolation scheme is used to calculate grid data for the granule. The latitude and longitude and associated grid row and column values are calculated for every 10th point in the GTM grid. Then quadratic interpolation is performed over 20x20 pixel rectangles within the GTM grid to obtain the full geolocation for the granule. The maximum error induced by the interpolation is one meter. Row times are calculated by performing linear interpolation between the granule boundary times.

2.3.7 Algorithm Support Considerations

DMS and INF must be running before execution of the GTM Imagery algorithm.

Currently, six of the 16 M-Band EDRs are created. The selection of which six bands to create is determined by an XML configuration file. The default bands are M1, M4, M9, M14, M15, and M16. It is the responsibility of the operator to select the appropriate bands for the mode of the granule (day, mixed, or night) that is to be processed.

The configuration file is in XML group name/config entry format. The group name describes the key value that is searched for by the GTM Imagery software to retrieve the corresponding config entries in the group. Each config entry consists of a name/configValue pair. There are configValue entries that describe whether the product is “Required” or “Optional” and one that sets the Wait flag to TRUE. All of the GTM keys described below must remain “Optional”, and “Wait” is set to TRUE in order to process cross granules. There are many other keys in the file, but only the name field (OfficialShortName_1) in the config entries in Table 25 below should be modified. The name of the XML configuration file is: /vobs/PRO/cfg/PRO_VIIRS_MChannelImagery_CFG.xml.

The key entries that need to be modified are described in Table 25.

Table 25. VIIRS M-Band Imagery Configuration

Group Name	Config name	Description	Default Value
VIIRS_GTM_Mod_Key_1	OfficialShortName_1	The key representing the M1ST product. Modify the configValue for the moderate band to be created for M1ST.	VIIRS-M1-FSDR
VIIRS_GTM_Mod_Key_2	OfficialShortName_1	The key representing the M2ND product. Modify the configValue for the moderate band to be created for M2ND.	VIIRS-M4-FSDR
VIIRS_GTM_Mod_Key_3	OfficialShortName_1	The key representing the M3RD product. Modify the configValue for the moderate band to be created for M3RD.	VIIRS-M9-FSDR
VIIRS_GTM_Mod_Key_4	OfficialShortName_1	The key representing the M4TH product. Modify the configValue for the moderate band to be created for M4TH.	VIIRS-M14-FSDR
VIIRS_GTM_Mod_Key_5	OfficialShortName_1	The key representing the M5TH product. Modify the configValue for the moderate band to be created for M5TH.	VIIRS-M15-FSDR
VIIRS_GTM_Mod_Key_6	OfficialShortName_1	The key representing the M6TH product. Modify the configValue for the moderate band to be created for M6TH.	VIIRS-M16-FSDR

2.3.8 Assumptions and Limitations

No assumptions or limitations have been identified.

3.0 GLOSSARY/ACRONYM LIST

3.1 Glossary

The current glossary for the NPOESS program, D35836_E_NPOESS_Glossary, can be found on eRooms. Table 26 contains those terms most applicable for this OAD.

Table 26. Glossary

Term	Description
Algorithm	A formula or set of steps for solving a particular problem. Algorithms can be expressed in any language, from natural languages like English to mathematical expressions to programming languages like FORTRAN. On NPOESS, an algorithm consists of: 1. A theoretical description (i.e., science/mathematical basis) 2. A computer implementation description (i.e., method of solution) 3. A computer implementation (i.e., code)
Algorithm Configuration Control Board (ACCB)	Interdisciplinary team of scientific and engineering personnel responsible for the approval and disposition of algorithm acceptance, verification, development and testing transitions. Chaired by the Algorithm Implementation Process Lead, members include representatives from IWPTB, Systems Engineering & Integration IPT, System Test IPT, and IDPS IPT.
Algorithm Verification	Science-grade software delivered by an algorithm provider is verified for compliance with data quality and timeliness requirements by Algorithm Team science personnel. This activity is nominally performed at the IWPTB facility. Delivered code is executed on compatible IWPTB computing platforms. Minor hosting modifications may be made to allow code execution. Optionally, verification may be performed at the Algorithm Provider's facility if warranted due to technical, schedule or cost considerations.
cm	Centimeter - unit of measurement for length.
EDR Algorithm	Scientific description and corresponding software and test data necessary to produce one or more environmental data records. The scientific computational basis for the production of each data record is described in an ATBD. At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Environmental Data Record (EDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to geophysical parameters (including ancillary parameters, e.g., cloud clear radiation, etc.). *[Supplementary Definition]* An Environmental Data Record (EDR) represents the state of the environment, and the related information needed to access and understand the record. Specifically, it is a set of related data items that describe one or more related estimated environmental parameters over a limited time-space range. The parameters are located by time and Earth coordinates. EDRs may have been resampled if they are created from multiple data sources with different sampling patterns. An EDR is created from one or more NPOESS SDRs or EDRs, plus ancillary environmental data provided by others. EDR metadata contains references to its processing history, spatial and temporal coverage, and quality.
IDPS Epoch Time (IET)	The standard for IDPS time storage. IET is the actual elapsed microsceconds, on the International Atomic Clock, based on an epoch date of 01 Jan 1958 (start of the International Geophysical Year. Also the base for TAI time)
K	Kelvin - unit of measurement for temperature.
M/s	Meters per second - unit of measurement for velocity.
Model Validation	The process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended uses of the model. [Ref.: DoDD 5000.59-DoD Modeling and Simulation Management]
Model Verification	The process of determining that a model implementation accurately represents the developer's conceptual description and specifications. [Ref.: DoDD 5000.59-DoD Modeling and Simulation Management]
Operational Code	Verified science-grade software, delivered by an algorithm provider and verified by IWPTB, is developed into operational-grade code by the IDPS IPT.

Term	Description
Operational-Grade Software	Code that produces data records compliant with the System Specification requirements for data quality and IDPS timeliness and operational infrastructure. The software is modular relative to the IDPS infrastructure and compliant with IDPS application programming interfaces (APIs) as specified for TDR/SDR or EDR code.
Raw Data Record (RDR)	*[IORD Definition]* Full resolution digital sensor data, time referenced, with absolute radiometric and geometric calibration coefficients appended, but not applied, to the data. Aggregates (sums or weighted averages) of detector samples are considered to be full resolution data if the aggregation is normally performed to meet resolution and other requirements. Sensor data shall be unprocessed with the following exceptions: time delay and integration (TDI), detector array non-uniformity correction (i.e., offset and responsivity equalization), and data compression are allowed. Lossy data compression is allowed only if the total measurement error is dominated by error sources other than the data compression algorithm. All calibration data will be retained and communicated to the ground without lossy compression. *[Supplementary Definition]* A Raw Data Record (RDR) is a logical grouping of raw data output by a sensor, and related information needed to process the record into an SDR or TDR. Specifically, it is a set of unmodified raw data (mission and housekeeping) produced by a sensor suite, one sensor, or a reasonable subset of a sensor (e.g., channel or channel group), over a specified, limited time range. Along with the sensor data, the RDR includes auxiliary data from other portions of NPOESS (space or ground) needed to recreate the sensor measurement, to correct the measurement for known distortions, and to locate the measurement in time and space, through subsequent processing. Metadata is associated with the sensor and auxiliary data to permit its effective use.
Retrieval Algorithm	A science-based algorithm used to 'retrieve' a set of environmental/geophysical parameters (EDR) from calibrated and geolocated sensor data (SDR). Synonym for EDR processing.
Science Algorithm	The theoretical description and a corresponding software implementation needed to produce an NPP/NPOESS data product (TDR, SDR or EDR). The former is described in an ATBD. The latter is typically developed for a research setting and characterized as "science-grade".
Science Algorithm Provider	Organization responsible for development and/or delivery of TDR/SDR or EDR algorithms associated with a given sensor.
Science-Grade Software	Code that produces data records in accordance with the science algorithm data quality requirements. This code, typically, has no software requirements for implementation language, targeted operating system, modularity, input and output data format or any other design discipline or assumed infrastructure.
SDR/TDR Algorithm	Scientific description and corresponding software and test data necessary to produce a Temperature Data Record and/or Sensor Data Record given a sensor's Raw Data Record. The scientific computational basis for the production of each data record is described in an Algorithm Theoretical Basis Document (ATBD). At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Sensor Data Record (SDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to calibrated brightness temperatures with associated ephemeris data. Temperature Data Records (TDRs) are geolocated, antenna temperatures with all relevant calibration data counts and ephemeris data to revert from T-sub-a into counts. The existence of the SDRs provides reversible data tracking back from the EDRs to the Raw data. *[Supplementary Definition]* A Sensor Data Record (SDR) is the recreated input to a sensor, and the related information needed to access and understand the record. Specifically, it is a set of incident flux estimates made by a sensor, over a limited time interval, with annotations that permit its effective use. The environmental flux estimates at the sensor aperture are corrected for sensor effects. The estimates are reported in physically meaningful units, usually in terms of an angular or spatial and temporal distribution at the sensor location, as a function of spectrum, polarization, or delay, and always at full resolution. When meaningful, the flux is also associated with the point on the Earth geoid from which it apparently originated. Also, when meaningful, the sensor flux is converted to an equivalent top-of-atmosphere (TOA) brightness. The associated metadata includes a record of the processing and sources from which the SDR was created, and other information needed to understand the data.

Term	Description
Tau	Unit of measurement for Optical Thickness.
Temperature Data Record (TDR)	*[IORD Definition]* Temperature Data Records (TDRs) are geolocated, antenna temperatures with all relevant calibration data counts and ephemeris data to revert from T-sub-a into counts. *[Supplementary Definition]* A Temperature Data Record (TDR) is the brightness temperature value measured by a microwave sensor, and the related information needed to access and understand the record. Specifically, it is a set of the corrected radiometric measurements made by an imaging microwave sensor, over a limited time range, with annotation that permits its effective use. A TDR is a partially-processed variant of an SDR. Instead of reporting the estimated microwave flux from a specified direction, it reports the observed antenna brightness temperature in that direction.
ViirsAmiIPType	VIIRS Aerosol Model Index Intermediate Product. Data is stored in an array of unsigned 8-bit integers.
ViirsAotIPType	VIIRS Aerosol Optical Thickness Intermediate Product. Data is stored in an array of 32-bit floating point numbers.
ViirsCloudMask IPType	A 48-bit word (6 bytes) for each moderate resolution pixel that includes information about whether the view of the surface is obstructed by clouds and specifies the processing path the algorithm took. Cloud phase data is also included as well as spatial uniformity, aerosol, shadow, and fire detection data.
ViirsIceConcIP Type	VIIRS Ice Concentration Intermediate Product. Data is stored in an array of 32-bit floating point numbers.
ViirsModBtType	VIIRS Moderate Resolution Channel Brightness Temperature. Data is stored in an array of 32-bit floating point numbers.
ViirsPwIPType	VIIRS Precipitable Water Intermediate Product. Data is stored in an array of unsigned 8-bit integers.
ViirsSnowIce CoverIPType	VIIRS Snow Ice Cover Intermediate Product. Data is stored in an array of 32-bit floating point numbers.
ViirsSelectSstLut Type	VIIRS Select SST Look Up Table.
ViirsSstCoeffsLut Type	VIIRS SST Coefficient Look Up Table.

3.2 Acronyms

The current acronym list for the NPOESS program, D35838_E_NPOESS_Acronyms, can be found on eRooms. Table 27 contains those terms most applicable for this OAD.

Table 27. Acronyms

Acronym	Description
AM&S	Algorithms, Models & Simulations
API	Application Programming Interfaces
ARP	Application Related Product
CDFCB-X	Common Data Format Control Book - External
DMS	Data Management Subsystem
DPIS ICD	Data Processor Inter-subsystem Interface Control Document
DQTT	Data Quality Test Table
IET	IDPS Epoch Time
IIS	Intelligence and Information Systems
INF	Infrastructure
ING	Ingest
IP	Intermediate Product
LUT	Look-Up Table
MDFCB	Mission Data Format Control Book
MDS	Map Data Set
NCC	Near Constant Contrast
PDL	Program Design Language
PRO	Processing
QF	Quality Flag
SDR	Sensor Data Records
SI	Software Item or International System of Units
SOM	Space Oblique Mercator (a conformal nearly equal area map)
TBD	To Be Determined
TBR	To Be Resolved
TOA	Top of the Atmosphere

4.0 OPEN ISSUES

Table 28. List of OAD TBD/TBR

No.	Description	Resolution Date
None		